NPS-52PW72041A

United States Naval Postgraduate School

HANDWRITTEN CHARACTER RECOGNITION

FROM PEN DIRECTION

by

V. MICHAEL POWERS

21 April 1972

NAVAL POSTGRADUATE SCHOOL
Monterey, California

Rear Admiral A. S. Goodfellow
Superintendent

M. U. Clauser
Provost

ABSTRACT:

A descriptive, generative model of the pen direction sequences measured during the hand drawing of characters is shown to be useful in recognition. Preprocessing and classification procedures are based on the properties of the model. Results are presented from tests with a realization of these methods as a real time recognition system running in a small computer.

This work was supported in part by the Foundation Research Program of the Naval Postgraduate School.

Chairman
Department of
Electrical Engineering

Dean of Research Administration

NPS-52PW72041A

21 April 1972

TABLE OF CONTENTS

Pen Direction Sequences in Character Recognition

V. Michael Powers

1. Introduction

Pattern recognition has been viewed as a problem of analysis by description. Several studies have addressed problems of description and recognition of line drawings. Hand-drawn characters have been used as examples of pattern classes; characters have been recognized by their description as a pattern of variously connected primitive elements [1]-[5].

This paper considers the problem of recognition of hand-drawn characters by means of on-line measurement of only one parameter of the drawing process: the short-term direction of the moving pen. Concentration on the direction parameter emphasizes the dynamic nature of the pattern production process. Processing techniques appropriate for economical on-line implementation are developed from a descriptive model of the data. This model not only includes a description of the pattern as a set of idealized, primitive elements but also allows for variation among patterns due to distortion and noise.

2. A Model

A generative model consisting of three sections, a grammar and two sets of transformations, is used as a hypothetical mechanism to generate strings similar to those measured during the drawing of a character. Each section produces a different level of description of the character. The problem of recognition can then be stated as an attempt to recover the highest-level description of a character drawing, its character class, from a lowest-level description a sequence of measured direction segments. Processing proceeds from the measurement string to an intermediate level to a classification decision.

Experimental measurements of direction were quantized to slopes consisting of the octants of a circle, numbered 0 through 7, beginning with the range 0° to 45° to the right of vertical. This set of slopes, plus markers corresponding to beginning and end of stroke (provided by a tip switch on the pen) are used as elements of the lowest-level description of a character.

The three components of the model are illustrated by the example in Figures 1, 2 and 3. A production of the generative grammar describes, for any one human subject, one of a small number of habitual ways of drawing a character. The description consists of a short sequence of character segments, punctuated by "pen down" and "pen up" marks. Figure 1 presents a production tree as it might exist in one person's writing vocabulary. The numeral 5 is shown as a sequence of actions: pen down (the circle), a downward

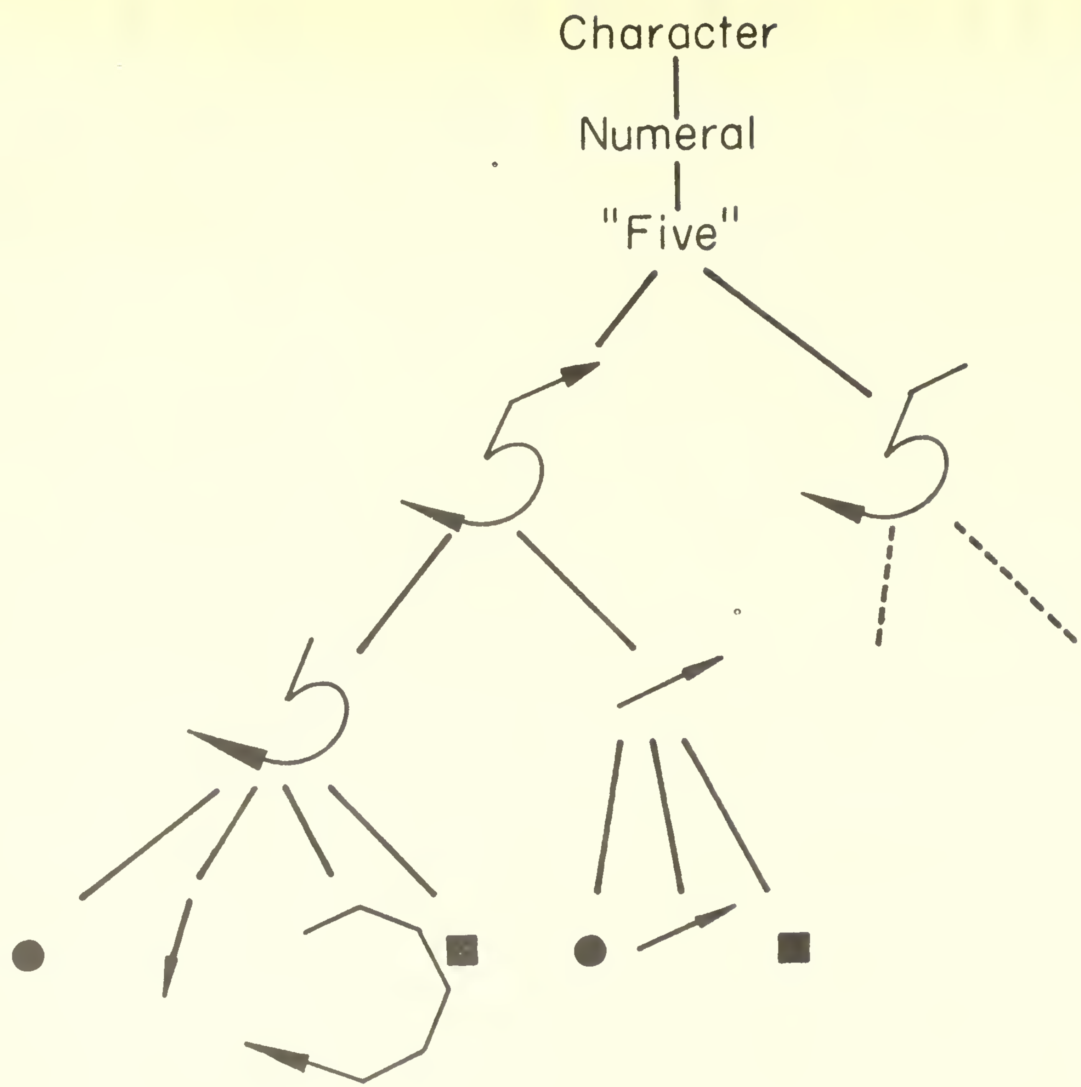
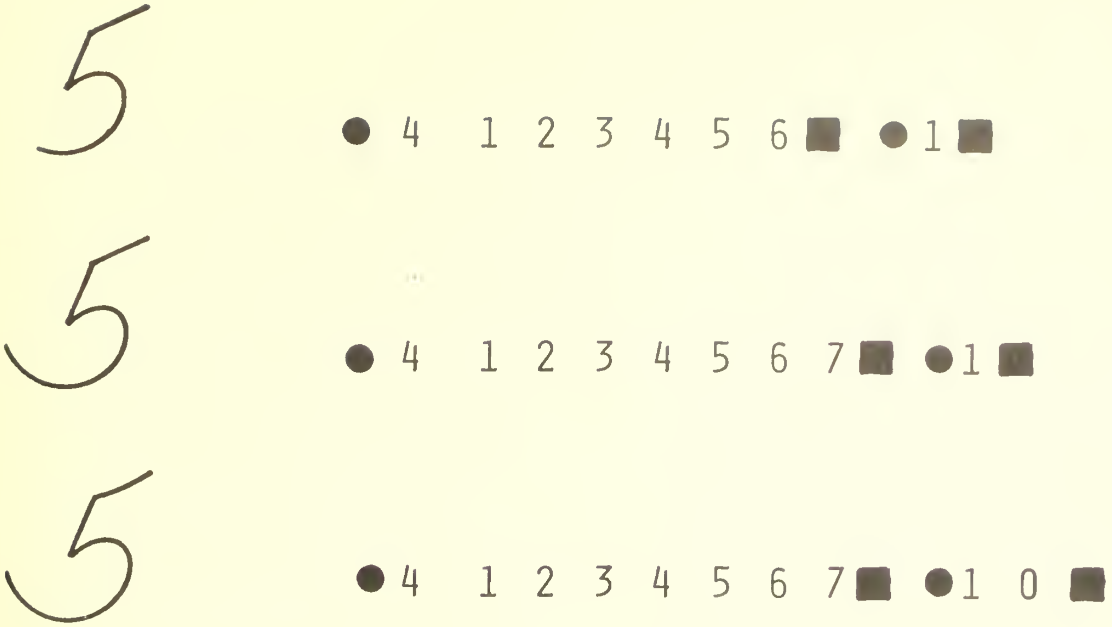


Figure 1. A Production of the Grammar

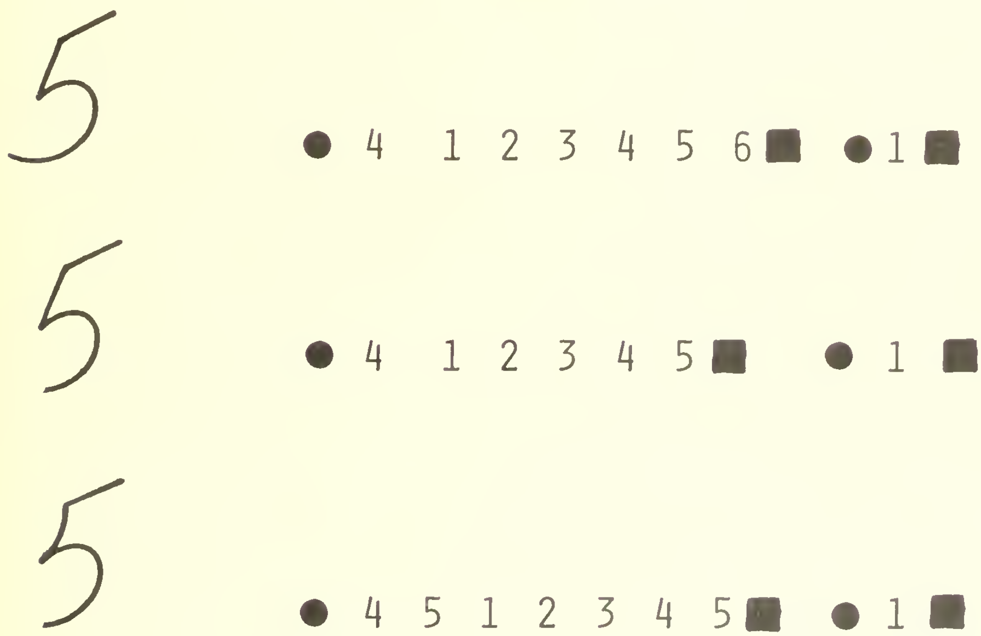
move, a clockwise curve, pen up (the square), pen down, a right-directed move, pen up. This string of primitive elements of a character is a sentence, or utterance, in the language of this grammar. Part of the production for a single-stroke drawing of the same character is shown on the right side of Figure 1. A person may have several alternate ways of drawing any single character.

Even within the specification given by the grammar, there are differences among repeated drawings of a single character. In our generative model, such deviations occur by means of the two remaining components of the model; two sets of transformations. The second model component is a set of smooth transformations and the third is a set of noise transformations. A transformation in either set maps among strings of the atomic elements: slopes and punctuation. Intuitively speaking, the smooth transformations account for the nonessential differences among repeated instances of the same character form; the noise transformations account for the degradation of the signal due to such noisy processes as hand tremor, pen drag, measurement and digitization.

Figure 2 shows examples of the effects of the smooth transformations. In Figure 2a, the 5 of Figure 1 is shown being successively distorted by overshooting the lower cusp and then curving up the tail of the upper bar. Figure 2b illustrates another pair of distortions which might appear as free variations in one person's writing: the lower cusp is shortened and the initial downstroke is curved.



2a) Extended Arcs



2b) A Curved Downstroke and An Abbreviated Arc

Figure 2. Examples from the Smooth Transformations

The second set, or noise transformations, are illustrated in Section 3b.

3. The Processing Space

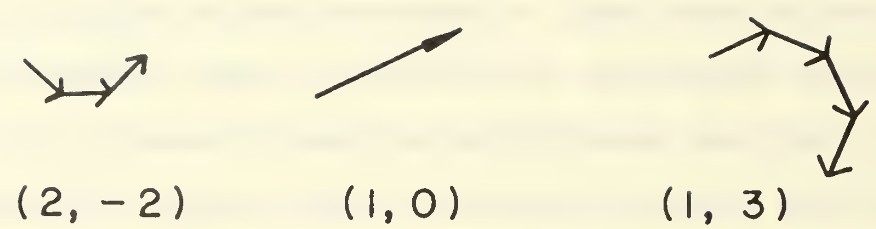
Separate productions by the three components of the model result in different direction sequences. The next section addresses the problem of trying to recognize direction sequences which are different but which represent the same characters. This section establishes a framework within which we can examine and quantify these differences. The results here are: a characterization of the direction sequences themselves in relation to their production by the model (and thus descriptive of their production by a human hand), and a pair of metrics which measure differences between two direction sequences which may have different lengths.

a. Arcs

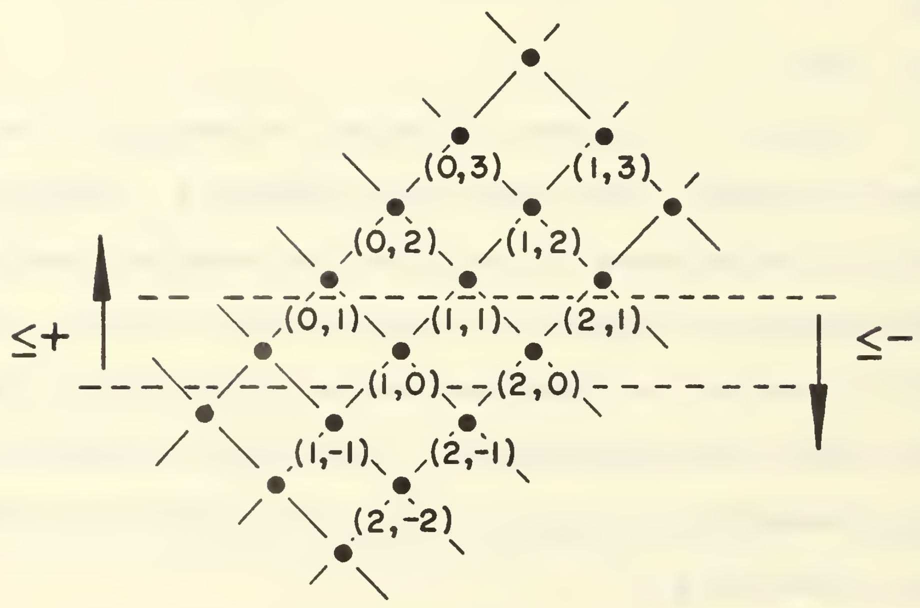
An arc is a smooth, monotone increasing or decreasing sequence of slopes. For any initial slope, s_i, an arc is defined by the sequence of positive (or negative) successors. Repeated slopes are ignored. Examples are: (2,1,0), (1), (1,2,3,4). An arc is characterized by its initial slope and the number of successors, n_i. The arc is clockwise or positive (negative) if n_i is a positive (negative) integer. We write, for arc a_i,

$$a_i = (s_i,n_i) = (s_i,\ldots,s_i \oplus n_i)$$

where $\oplus$ of course denotes addition modulo 8. It is convenient to use mixed notation, where $s_i \varepsilon\{0,1,2,3,4,5,6,7\}$ and n_i is a signed integer. The three examples above are thus written: (2,-2),(1,0),(1,3) as illustrated in Figure 3a.



3a) Three Arcs



3b) An illustration of the relation $\leq$

Figure 3. Arcs

The generative grammar produces a sequence of arcs; the smooth transformations extend and truncate them. Processing and identification of arcs must be based on the relation among similar arcs. For arcs of the same sign, we wish to use the partial order relation defined by the inclusion of one slope sequence in another. Two natural partial order relations thus exist. They will be used only in the subsequent definition.

For arcs $a_i = (s_i, n_i)$ and $a_j = (s_j, n_j)$ we define:

$a_i \leq_+ a_j$ if n_i and n_j are ≥ 0, and (the slope sequence) a_i is a subsequence of a_j.

$a_i \leq_- a_j$ if n_i and n_j are ≤ 0 and a_i is a subsequence of a_j.

For example, $(3,1) \leq_+ (3,2)$, $(3,1) \leq_+ (2,2)$, $(6,2) \leq_- (7,-4) \underline{-} (0,-6)$, $(2,0) \leq_+ (2,1)$, $(2,0) \leq_- (2,-1)$.

This pair of relations is not sufficient, however. It does relate the arcs representing similar curves of either rotational sense, but does not relate positive and negative arcs. Noting that the arcs of straight lines $(s_i, 0)$ are the lesser arcs in either relation, we define a relation on all arcs by the transitive extension through these "simplest" members. For a_i and a_j as before, we define:

$a_i \leq a_j$ if a) $a_i \leq_+ a_j$

or b) $a_i \geq_- a_j$

or c) There exists an arc, a_k, such that

$a_i \geq_- a_k \leq_+ a_j$.

Note that for simplicity in this representation we turn one of the defining relations (arbitrarily $\leq_-$) "upside down". Figure 3b illustrates the relationships among the arcs of Figure 3a. We have (1,0) $\leq_+$ (1,3) and (1,0) $\leq_-$ (2,-2) so that (2,-2) $\leq$ (1,0) $\leq$ (1,3).

A larger portion of the space thus developed is shown in Figure 4. There several arcs are shown as points, and a few are labelled. In the upper portion, for example, are positive arcs with (0,5) $\leq$ (0,6) $\leq$ (0,7) $\leq$ (7,8). Negative arcs are in the lower portion, and near the middle are the eight straight lines $(n_i,0)$. A straight line is thus merely a special case of an arc. This is appropriate for hand-drawn characters where repeated samples of a nominally straight line may bow slightly in either direction. Thus the downward-left slant (4,0) may become a clockwise curve((4,1) or (3,1)) or a counterclockwise curve ((4,-1) or (5,-1)).

The space as defined can be thought of as cylindrical. For example in Figure 4, the arc to the "right" of (7,0) is (0,0), and (7,0) $\leq$ (7,1) $\leq$ (7,2). An arbitrary pair of arcs has at least one upper and one lower bound, but not necessarily a least upper bound or greatest lower bound. The pair of arcs (0,0) and (4,0) have upper bounds (0,4) and (4,4) but these bounds are not related. A subset of arcs which includes both an l.u.b. and a g.l.b. and all the arcs "between" is an <u>arc lattice</u>. An arc lattice will be uniquely represented by its two extremes.

Although the arc space as defined is not bounded above or below, the applications to follow only use a finite portion. The range $(s_i,16)$ to $(s_i,-16)$ will suffice.

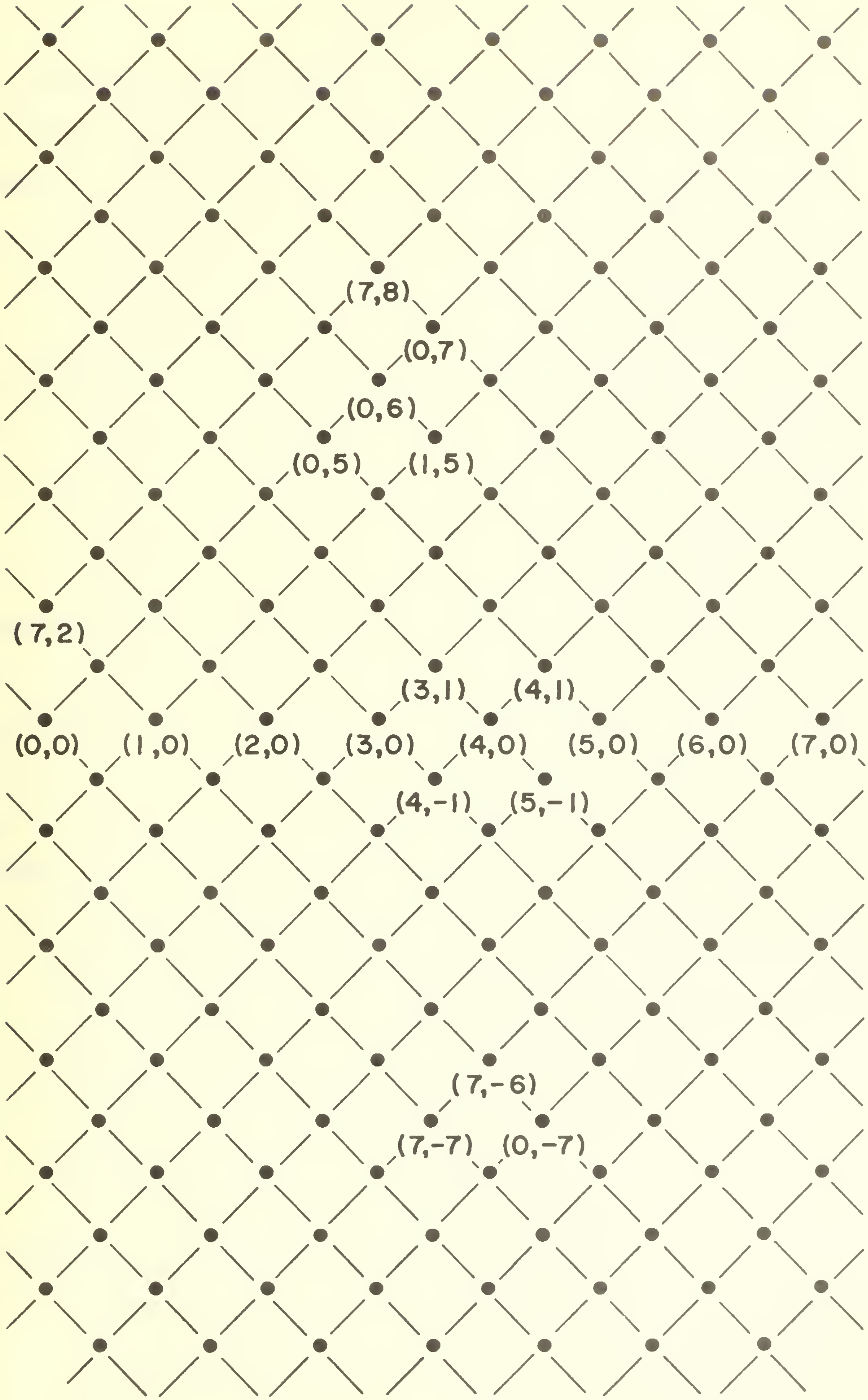


Figure 4. The Arc Space Ordered by $\leq$.

There is a simple test which characterizes the ordering. Let $s_i \ominus s_j$ be the difference modulo 8 of the two slopes. Then for any two arcs $a_i = (s_i,n_i)$ and $a_j = (s_j,n_j)$ it can be shown that

$$a_i \leq a_j \iff n_j - n_i \geq s_i \ominus s_j \quad .$$

More importantly, for any arc lattice, a, denoted by its l.u.b. a_j and g.l.b. a_k, and written

$$a = \begin{vmatrix} a_j \\ a_k \end{vmatrix}$$

we now have a simple test for the inclusion of any arc a_i in the lattice:

$$a_i \varepsilon\ a \iff a_i \leq a_j \text{ and } a_k \leq a_i$$

b) Distortions and metrics

The smooth transformations of the generative model account for distortions of the general shapes of the arcs. The structure of the arc lattice, now, provides a basis for portraying small distortions as near neighbors. In particular, the immediate neighbors of an arc (s_i,n_i) are $(s_i,n_i + 1)$, $(s_i,n_i - 1)$, $(s_i \oplus 1, n_i - 1)$, $(s_i \ominus 1, n_i + 1)$ - the four arcs formed by appending or deleting strobes at (curling or straightening) either end of the arc.

The set of smooth transformations could then be cast as a distribution of changes which introduce different amounts of distortion, or as a sequence of elementary transformations, each of which distorts the arc by one

"step" to a nearest neighbor. A detailed investigation of the model mechanics is not warranted at this point, however. We shall merely characterize the overall distortion effect of this stage of the model.

We define a measure of the difference between two arcs (of different sequence length, in general) corresponding to their distance in number of steps on the arc lattice. The value of this <u>arc metric</u>, d_a, between two arcs a_i, a_j, is the length of a minimal path between a_i and a_j along pairwise related arcs.

The last component of the model, the noise transformations, provide another sort of difference among slope sequences. Quantization error, instrumentation noise, writing surface roughness or hand tremor can produce erratic disturbances in an otherwise smooth stream of slopes. For example, suppose the arc (4,-3) were produced by the grammar and smooth transformation, and corresponded to the sequence of slopes:

4,4,4,4,4,4,3,3,2,1,1,1

then the following list might correspond to the influence of successive minor noisy aberrations:

4,4,4,4,4,4,3,3,2,1,1,2,1

4,4,3,4,4,4,4,3,3,2,1,1,2,1

1,4,4,3,4,4,4,4,3,3,2,1,1,2,1 .

The effect of such noise on the description in terms of arcs is disconcerting. This simple curve becomes

(1,0), (4,-1), (4,-3), (2,-1) .

In order to deal with this spurious segmentation of a smooth curve by noise, we need a measure of this effect.

We define a second metric, a noise metric, d_n, on sequences of slopes. The distance between two slope sequences will again be the minimum length of a sequence of small changes. Contrasting with the arc metric above, however, the minimal change or step in this chain will be the insertion or deletion of an arbitrary slope. Thus the distance between the first and fourth slope sequence above, as measured by the noise metric is three.

4. Processing Methods

A goal in recognition is to distinguish among various pattern classes. Specifically, in terms of our model this goal becomes one of distinguishing among different productions of the grammar, given a direction sequence which has been distorted by both smooth and noise transformations. In this parsing attempt we shall be concerned with simple methods for preprocessing and classifying sample slope sequences.

a) Preprocessing by Compaction

The primary effect of the noise transformations mentioned above is to break smooth arcs into a multitude of short segments. The amount of damage, in terms of distortion, can be estimated by the number of segments introduced. A preprocessing or filtering of the noisy data should recover many of the smooth arcs from the splintered remains. Any slope sequence can be transformed to any other by a long enough path of noise transformations; what is desired is that we protect against disturbances due to small (by the noise metric) amounts of noise.

The preprocessing step is an attempt to recover a smooth slope sequence (a small number of arcs) from a noisy one by repeated adjustments. The constraint, which prevents filtering of significant changes, is expressed in the following <u>compaction principle</u>:

> Any change of distance $d_n = 1$ is allowed which reduces the number of arcs by at least 1.

b) Classification

Given a preprocessed version of a signal representing a pattern, we wish to classify the signal according to which one of the possible patterns it most likely represents. In terms of parsing a production of our model, we start with an arc sequence - a distorted version of a character - and we attempt to decide which of the characters in our vocabulary corresponds to that arc sequence.

In a realistic pattern recognition situation, there are a number of difficulties in making this choice. A realistic source of patterns generates ambiguous signals - signals which could be representative of more than one pattern type. Making the best choice in classifying ambiguous signals depends on knowledge of the probability distributions of the respective signal classes, but this knowledge is usually difficult to obtain without very extensive testing.

Fortunately, relatively simple assumptions lead to a very convenient solution in some cases. If each pattern is equally likely to occur, and if the distribution of samples is the same function of distance for each pattern, then a received sample is most likely a version of the nearest pattern. If only a collection of correctly classified samples is available, and not the complete description of the distributions, then it is known that classifying a new sample according to its nearest neighbors is a reasonable choice in terms of probability of error [6].

Maintaining an exhaustive list of samples and their classifications is a tiresome task, however. A convenient

approximation to this information is provided by a mechanism suggested earlier in this paper. Known samples of arcs are collected into arc lattices. These arc lattices can be stored very compactly (each is represented by an l.u.b. and a g.l.b.). Classification of an arc sample then becomes a question of determining membership in a characterizing arc lattice. This test of membership is an extremely simple calculation, as shown earlier.

One intuitively satisfying benefit accrues from this approximation. If somehow a sample appears which is far from any previously classified sample we avoid merely assigning it to the most likely class. The principle effect of this "don't know" classification is to point out to a system with only finite experience (limited learning) both samples of new, previously unseen patterns and samples which are unexpected distortions of "known" patterns.

5. An Example Application

A system has been constructed for experimentation with the approach described above. In keeping with the spirit of economy implied by the use of only a single parameter of the characters' production, the time-ordered direction sequence, a modest amount of hardware was employed.

a. Hardware

The input device is an experimental modification of an Electrowriter® terminal. It provides signals roughly proportional to the horizontal and vertical coordinates of a ball point pen within a small writing area (about 3in. by 5in.), and terminals to a microswitch which is closed when the pen is pressed to the writing surface.

A small analog computer scales the position coordinates and forms "pen up" and "pen down" signals from the pen switch actions.

A small digital computer, a LINK-8, converts the analog signals and accomplishes all further processing. Except for sampling instructions and a section to escape into a magnetic tape loader, the PDP-8 processor (rather than the LINK processor) is used exclusively, with program and data in the 4K(12-bit) core memory.

b. Software

A multiprogramming system structure modified from a data concentrator application [7] performs utility functions and manages the processing in several concurrent streams or process tasks.

The direction sequence is approximated by quantizing

the slopes between successive points. A point is the first sampled position pair to differ from the last point (by more than a fixed distance (empirically determined). Position sampling and slop extraction is performed continually whenever the pen is down. Intervals between samples vary depending on the execution time required by other tasks. After the slope quantization, relative and absolute position are discarded. Further processing uses only the sequence of measured slopes, punctuated by stroke delimiters "pen down" and "pen up".

A major portion of the program size is devoted to a preprocessing section which implements the compaction principle discussed earlier. A tree structure of tests changes the slope sequences by a noise metric distance of one whenever such a change can reduce the number of smooth sequence segments, i.e., arcs.

Because of the distance limitation on the amount of allowable smoothing, arcs early in the order of production are often passed through this preprocessing task before the stroke ends. Thus the initial steps in classification of a character can begin before the character is finished. This concurrency of processing provided by the multiprogramming system allows effectively immediate response by the system. Although the final classification response is postponed until the character is finished, no delay in output is noticeable when the machine types its response.

c. System operation and application.

The system has been used in attempts to recognize drawings of the Arabic numerals. This alphabet was chosen because it is small, it includes many differences between different people, and it includes an inherent ambiguity. For most people's vocabularies, the 6 and the 0 have similar direction sequences, and slight overshoot and undershoot at the beginning and end of the stroke change one into the other. In terms of arcs, the arc representing a standard or carefully drawn 6 is very close to that for an 0, and the distributions representing repeated drawings of the two overlap considerably.

One way to resolve the ambiguity is to examine parameters of the character other than the direction sequence. For example, the relative distance between the start and end point of the character would seem to be a reliable discriminant between 6 and 0. If a relatively simple approach such as ours can be used in conjunction with measurement of other parameters, then this ambiguity as well as most of the other errors can be corrected. Perhaps, as suggested by others [8], a quick simple system such as this should be used as a "front end", classifying the samples which it can confidently identify, and passing questionable samples to another system; or perhaps several systems should operate in parallel.

Another way to resolve the ambiguity is to change the input patterns to make them distinguishable. The author, while developing the system, quite painlessly adopted a style in his own writing which consistently discriminates

between the two. It was found that a simple instruction to users was sufficient to enable them to produce descriminable 6s and 0s. One technique was to add a simple loop at the finish of a zero.

All recognition results reported below were performed under a rather stringent structuring of the classification section. First, no pattern was identified unless every arc of every stroke of the character agreed with the description in the prepared dictionary (a list of arc lattices describing permissible distortions of each arc). No "partial credit" was allowed for slope sequences which agreed in most sections. Second, an unambiguous decision was demanded. In some applications, a partial classification (i.e. "6 or 0") would be better than a wrong one. Here, the result of a forced choice was sometimes wrong.

The classification procedure is a sequential match of sample arcs against dictionary entries composed of strings of arc lattices. The classification decision, therefore, is embodied in the construction of the dictionary. In all cases to date, this dictionary has been constructed by hand. Heuristic techniques for dictionary construction have been developed, but the scope and volume of data have not yet justified mechanizing the process. The most promising procedure seems to be to start with a dictionary and modify it for an individual on the basis of recognition errors. This initial dictionary might be a compilation of samples, but in practice it seems to be more convenient to start with a dictionary constructed from visual examination of a single

sample of the subject's writing, or to start with another person's dictionary. In updating a given dictionary for a given subject, the system is directed to print out intermediate processing results, such as smoothed arc sequences, while trying to classify inputs with the old dictionary. The occasional to frequent successful recognition during this data-gathering run have been observed to act as satisfying, positive reinforcement for the subject. The results of such a run are used to update the dictionary; arc lattice boundaries are changed in an attempt to simultaneously minimize the number of samples not identified, and the number identified wrong. A major constraint during this procedure is that the dictionary be unambigious; no arc sequence may match more than one entry.

Results of five recognition tests are summarized in Table 1. [9]. Each column presents a separate test. N samples were taken, divided almost equally among the ten numerals. The percentages of samples correctly identified (A), not identified (X), and incorrectly identified (M) are shown. Columns (a) and (b) represent tests on different days with the same subject and the same dictionary; he was not instructed to modify his method of drawing 6 or 0. His habits may have been modified by the experience of immediately seeing the result of each classification attempt, however. Columns (c) and (d) are tests with a subject who has learned to distinguish his 6 and 0 articulations. Column (e) is a test with an important difference. Here, the dictionary from tests (c) and (d) was used without modification, after it

	(a) DLHa	(b) DLHb	(c) VMPa	(d) VMPb	(e) CRP
N	233	250	250	108	231
A/N	79.4%	72.8%	90.8%	92.8%	90.2%
X/N	16.1%	21.2%	9.2%	7.2%	8.6%
M/N	4.7%	6%	0	0	1.3%

Table 1. Summary of Test Response Categories

was demonstrated to him. His 3 errors (1.3%) were 6's identified by the system as 0's.

6. Conclusion

It has been shown that the application of the arc distance concept can lead to a character recognition system which operates in a small machine and provides real time decisions.

Intentionally limiting the measurements to a time-ordered sequence of approximations to pen direction has two important implications. First, this parameter is size-independent. Matching samples against arc lattices allows considerable tolerance of variations which appear during repeated writing of a given character, including variation in the overall slant. These characteristics make this approach especially attractive for use with a pen-direction sensor in cases where the human cannot or should not have to pay particular attention to the appearance of his writing - he might even write over previous characters.

Second, use of this single parameter of character production may involve irresolvable ambigiuties when applied to a conventional alphabet. These ambiguities can be resolved by changing the alphabet, however. Either the conventional alphabet can be modified, or an artificial alphabet can be constructed for special applications. Another approach would be to use this direction sequence parameter in conjunction with another parameter, such as relative position.

Several steps would be required before a system based on the principles described here could compete with present methods, such as optical character recognition, in applications. The design of a pen motion transducer is not discussed here; our experiments have only simulated its readings. In any

further extension, more attention should be paid to the preprocessing (filtering) stage; in most cases when the sample, it was because the sample was improperly segmented into component arcs.

References

[1] A. Guzman-Arenas, "Computer recognition of three-dimensional objects in a visual scene," Project MAC, MAC-TR-59, M.I.T., Dec. 1968.

[2] M. Eden and P. Mermelstein, "A system for automatic recognition of handwritten words", AFIPS Proc. FJCC, vol. 26, part I, 1964, pp 333-342.

[3] W. F. Miller and A. C. Shaw, "Lingiustic methods in picture processing - a survey", AFIPS Proc FJCC, vol. 33, part I, 1968, pp 279-290.

[4] R. Narasimhan and V.S.N. Reddy, "A generative model for handprinted English letters and its computer implementation", ICC Bulletin, vol. 6, no. 4, Oct-Dec. 1967, pp 275-287.

[5] A. C. Shaw, "The formal description and parsing of pictures", Stanford University Ph.D. thesis and Report SLAC-84, VC-32, Math and Computers, 1968.

[6] T. M. Cover and P. E. Hart, "Nearest neighbor pattern classification," IEEE Trans. Inf. Th., vol IT-13, Jan. 1967, pp 21-27.

[7] D. L. Mills, "RAMP: a multiprogramming system for real-time device control," Concomp Project Memo 5, University of Michigan, May 1967.

[8] G. Caluimi, "Un metodo linguistico per il riconoscimento automatico in tempo reale di caratteri scritti a mano", CISE-120, CISE, Milan, Dec. 1968.

[9] V. M. Powers, "On-line recognition of hand-drawn characters from directional information", Ph.D. thesis, The University of Michigan, 1970.

DISTRIBUTION LIST

Defense Documentation Center Cameron Station Alexandria, Virginia	12
Library Naval Postgraduate School Monterey, California 93940	2
Dean of Research Administration Naval Postgraduate School Monterey, California 93940	2
Professor V. Michael Powers Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	8
Professor Sydney R. Parker Chairman, Dept. of Electrical Engineering Naval Postgraduate School Monterey, California 93940	2

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Naval Postgraduate School Monterey, California	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP

3. REPORT TITLE

Handwritten Character Recognition from Pen Direction

4. DESCRIPTIVE NOTES (Type of report and, inclusive dates)
Technical Report, 1971-72

5. AUTHOR(S) (First name, middle initial, last name)
V. Michael Powers

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
21 April 1972	32	9

8a. CONTRACT OR GRANT NO. Foundation Research Program	9a. ORIGINATOR'S REPORT NUMBER(S) NPS-52PW72041A
b. PROJECT NO.	
c.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
d.	

10. DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY

13. ABSTRACT

A descriptive, generative model of the pen direction sequences measured during the hand drawing of characters is shown to be useful in recognition. Preprocessing and classification procedures are based on the properties of the model. Results are presented from tests with a realization of these methods as a real time recognition system running in a small computer.

This work was supported in part by the Foundation Research Program of the Naval Postgraduate School.

DD FORM 1 NOV 65 1473 (PAGE 1)

S/N 0101-807-6811

UNCLASSIFIED
Security Classification

A-31408

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Character Recognition						
Hand-drawn Characters						
Real Time Recognition						
Handwriting Descriptive Model						
Pen Direction Sequences						

DD FORM 1 NOV 65 1473 (BACK)

S/N 0101-807-6821

UNCLASSIFIED

Security Classification

A-31409